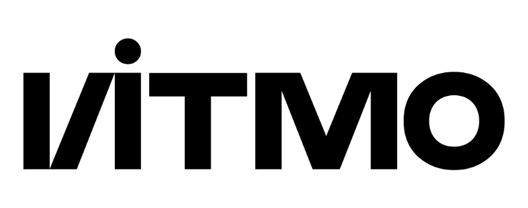
Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Домашнее задание #1

по дисциплине “Системы ввода-вывода”

Вариант: 3

**Выполнили:**

Разинкин Александр Владимирович 1.1

Щербинин Эдуард Павлович 1.1

Сиразетдинов Азат Ниязович 1.1

Баянов Равиль Динарович 1.4

Хабнер Георгий Евгеньевич 1.5

**Преподаватель:**

Быковский Сергей Вячеславович

г. Санкт-Петербург

2025

**Текст задания**

Спроектировать интерфейс передачи данных, удовлетворяющий следующим требованиям:

* Количество линий: 6
* Синхронный
* Дуплексный

**Этап 1. Проектирование портов ввода/вывода**

Распиновка разъема:

* MDO1 (Master Data Out 1) - передача первого бита от master-устройства
* MDO2 (Master Data Out 2) - передача второго бита от master-устройства
* MDI1 (Master Data In 1) - передача первого бита от slave-устройства
* MDI2 (Master Data In 2) - передача второго бита от slave-устройства
* CLK - передача сигнала синхронизации
* - передача сигнала выбора первого slave-устройства
* - передача сигнала выбора второго slave-устройства
* GND - “земля”

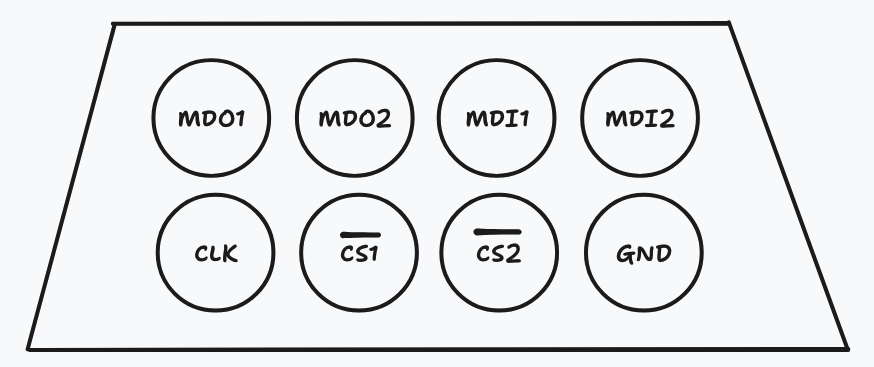
Как можно заметить, данный интерфейс имеет по две пары линий как для вывода (MDO1/2), так и ввода (MDI1/2) master-устройства. Благодаря данному решению передача данных осуществляется в два раза быстрее, так как биты передаются парами. Например, при передаче байта от master-устройства биты будут передаваться одновременно по линиям MDO1 и MDO2 в следующей последовательности (приведены индексы битов):

MDO1: 0 2 4 6

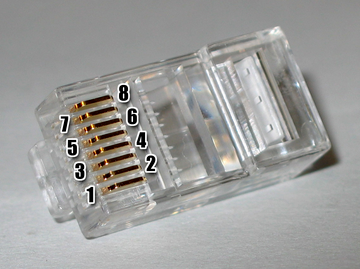
MDO2: 1 3 5 7

В качестве примера был выбран интерфейс, поддерживающий два slave-устройства, но можно увеличить их количество путем увеличения количества линий CS.

Примерный вид разъема в разрезе для случая для выбранного примера:



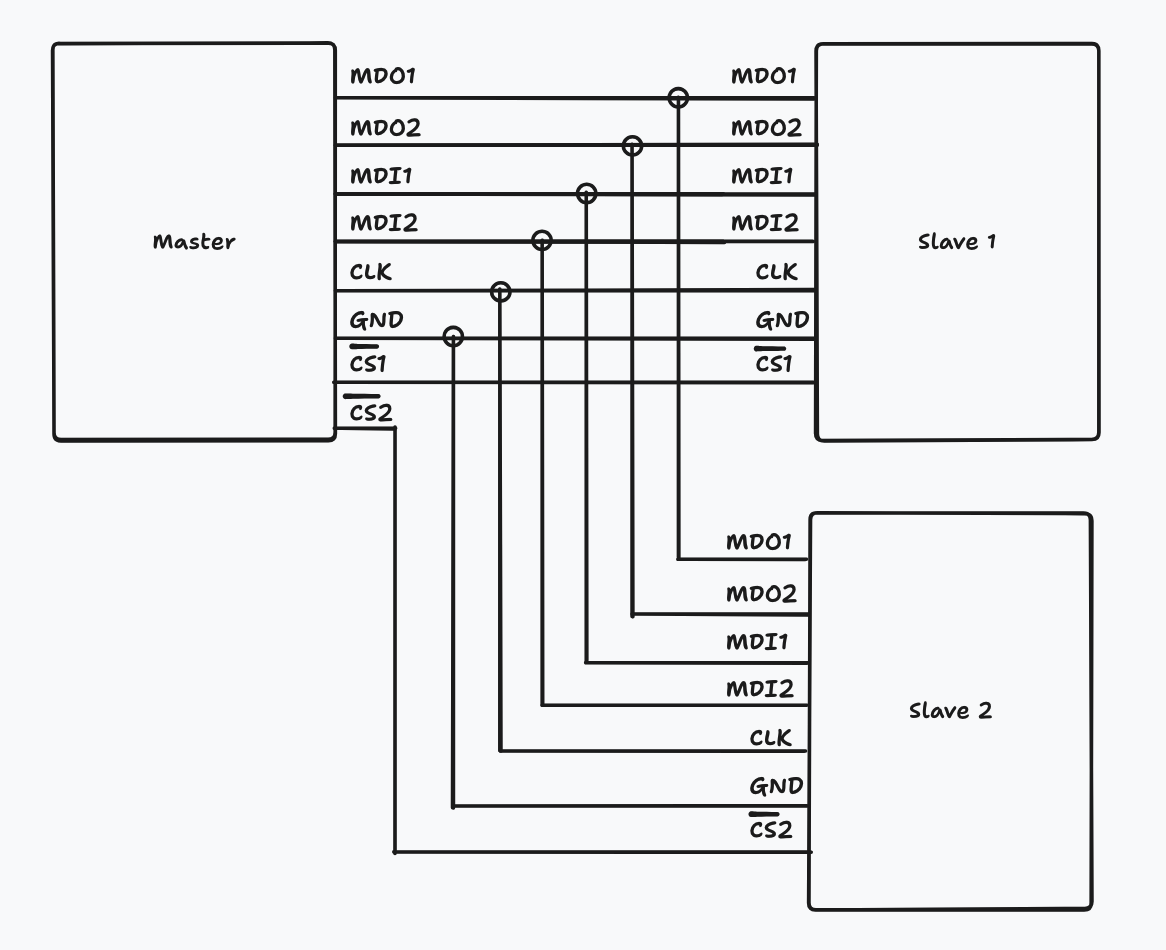
В качестве существующего разъема можно взять 8P8C:



Возможные виды подключения для выбранного примера:

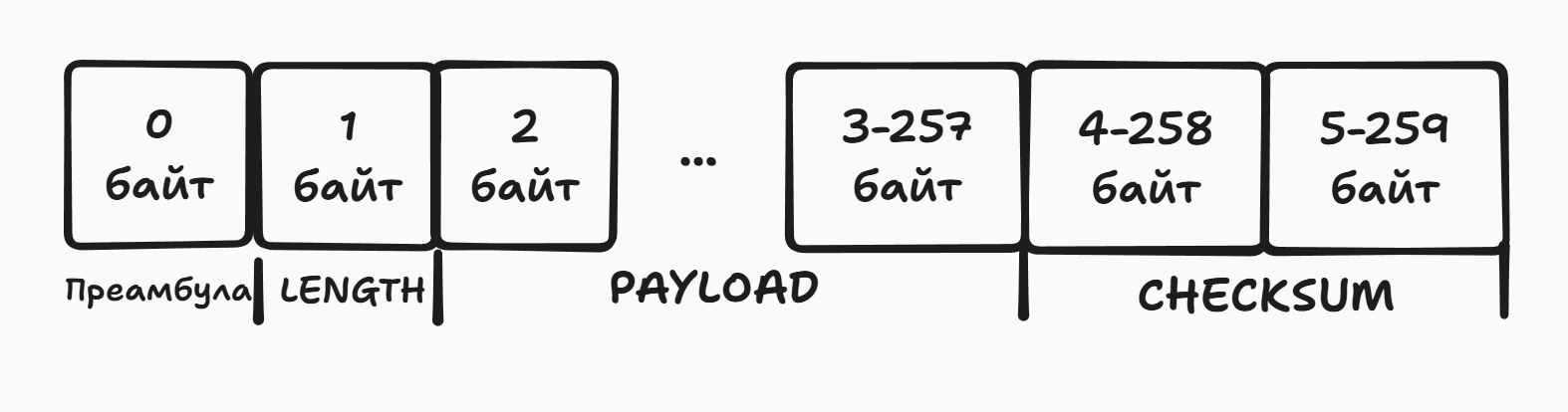
* Звезда - подключены два slave-устройства
* Один-к-одному - подключено одно slave-устройство

Схематическое представление подключения для случая с двумя slave-устройствами:



**Этап 2. Проектирование протокола передачи данных**

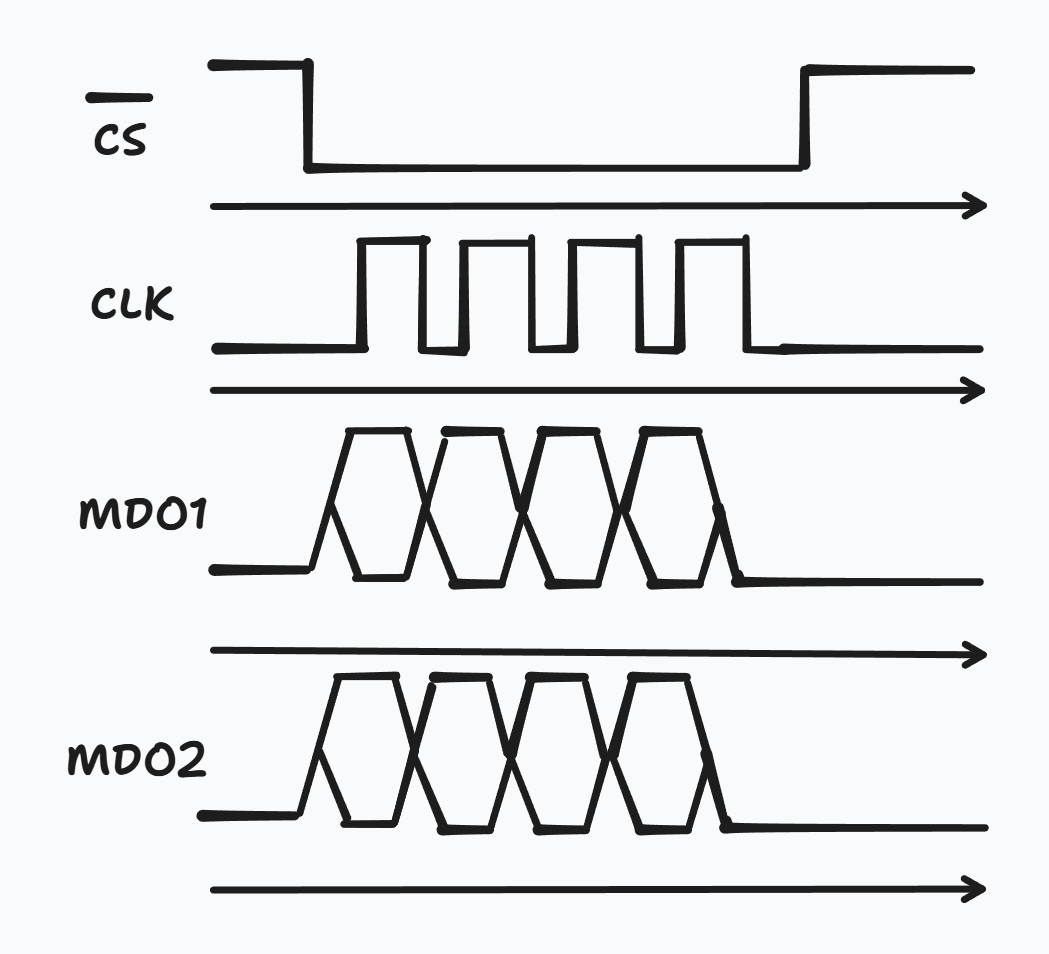
Формат пакета канального уровня:



Описание пакета:

* Преамбула - 1 байт: магическое число (10101010), являющееся сигналом от устройства к устройству о том, что оно хочет передать пакет (см. Этап 3).
* Длина пакета (LENGTH) - 1 байт: длина (в байтах) полезной нагрузки PAYLOAD, следующей далее.
* Полезная нагрузка (PAYLOAD) - 1-256 байт: полезные данные.
* Контрольная сумма (CHECKSUM) - 2 байта: сумма всех байтов полезной нагрузки, деленная по модулю на 2^16.

Один байт данных нагрузки передается по линиям следующим образом:



* Master адресует устройство, к которому желает обратиться, установив “0” на соответствующей линии Chip Select конкретного устройства, что также служит сигналом к началу передачи данных
* По линии синхронизации Clock Master передает Slave сигнал синхронизации, на фронте которого будет зафиксирован каждый переданный бит.
* По линиям MDO1 и MDO2 одновременно и параллельно передаются по 4 бита одного байта, что дает двукратный прирост физической скорости передачи, и один байт передается за 4 сигнала синхронизации.
* Таким способом передается преамбула, и если устройство готово принять пакет (см. Этап 3), передается оставшаяся часть пакета. Получившее пакет устройство сравнивает контрольную сумму, и отправляет результат. Если результат положительный, устройство переходит к передаче следующего пакета, иначе, текущий пакет отбрасывается и отправляется заново.

Рассчитаем полезную скорость передачи данных, с учетом скорости линии 1 Мбит/с и максимальной загрузки пакета, с полезной нагрузкой в 256 байт.

* Коэффициент полезных данных: 256/260 байт
* Скорость передачи полезных данных по одной линии:  
  1 Мбит/с \* 256/260 = 0.985 Мбит/с
* Общая пропускная способность интерфейса (по двум линиям):  
  0.985 Мбит/с \* 2 = **1.97 Мбит/с**

**Этап 3. Описание сценариев использования и протокола транспортного уровня**

**Сценарии использования и прикладные области:**

Может быть полезно в системах реального времени, где важна скорость, безопасность и надёжность передачи данных на короткие расстояние (между датчиками и разными электронными устройствами).

Например:

* Автомобильная электроника: Нашим мастер-устройством может выступать блок-управления автомобилем, а slave будут представлять в виде различных датчиков температуры, давления в шинах, датчики скорости, положения руля и положения объектов, рядом стоящих с автомобилем. В автомобиле крайне важна тормозная система ABS, которая будет ярким примером трафика реального времени, так как важно с огромной частотой, осуществлять проверку состояния колёс.
* Медицинское оборудование: Slave могут представляться в виде различных датчиков, определяющих состояния здоровья человека. Это также пример трафика реального времени, так как скорость и точность, предоставляемых данных, может сыграть значительную роль в жизни человека.

Данные примеры достаточно хорошо реализуемы нашим интерфейсом, благодаря высокой пропускной способности, за счет сдвоенных портов ввода-вывода, а также за счет синхронизированной передачи данных, что обеспечивает точность и надёжность передачи информации.

Для обеспечения надежности используются следующие методы:

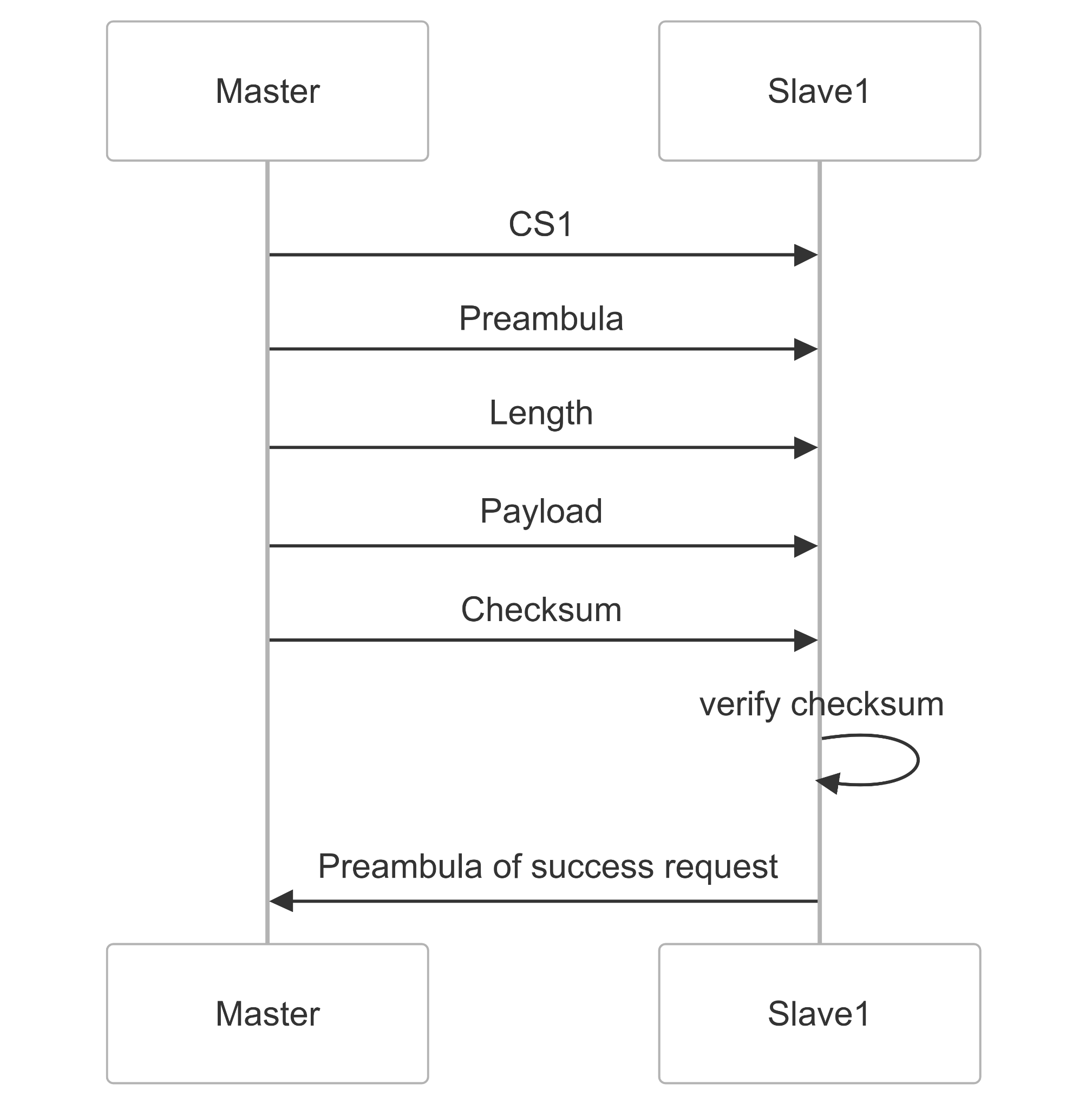
* В каждом пакете передается 2-байтная контрольная сумма для проверки корректности переданного сообщения.
* Если после таймаута отправитель не получает подтверждение о корректности подсчитанной контрольной суммы, то пакет передается заново.

Протокол на транспортном уровне:

* Отправка пакета устройством.
* Получение значения контрольной суммы.
* При отсутствии ответа повторная отправка.

Организация обмена данными между Master и Slave

1. Передача данных от Master к Slave:



- Master выбирает целевое Slave-устройство с помощью сигнала CS (Chip Select).

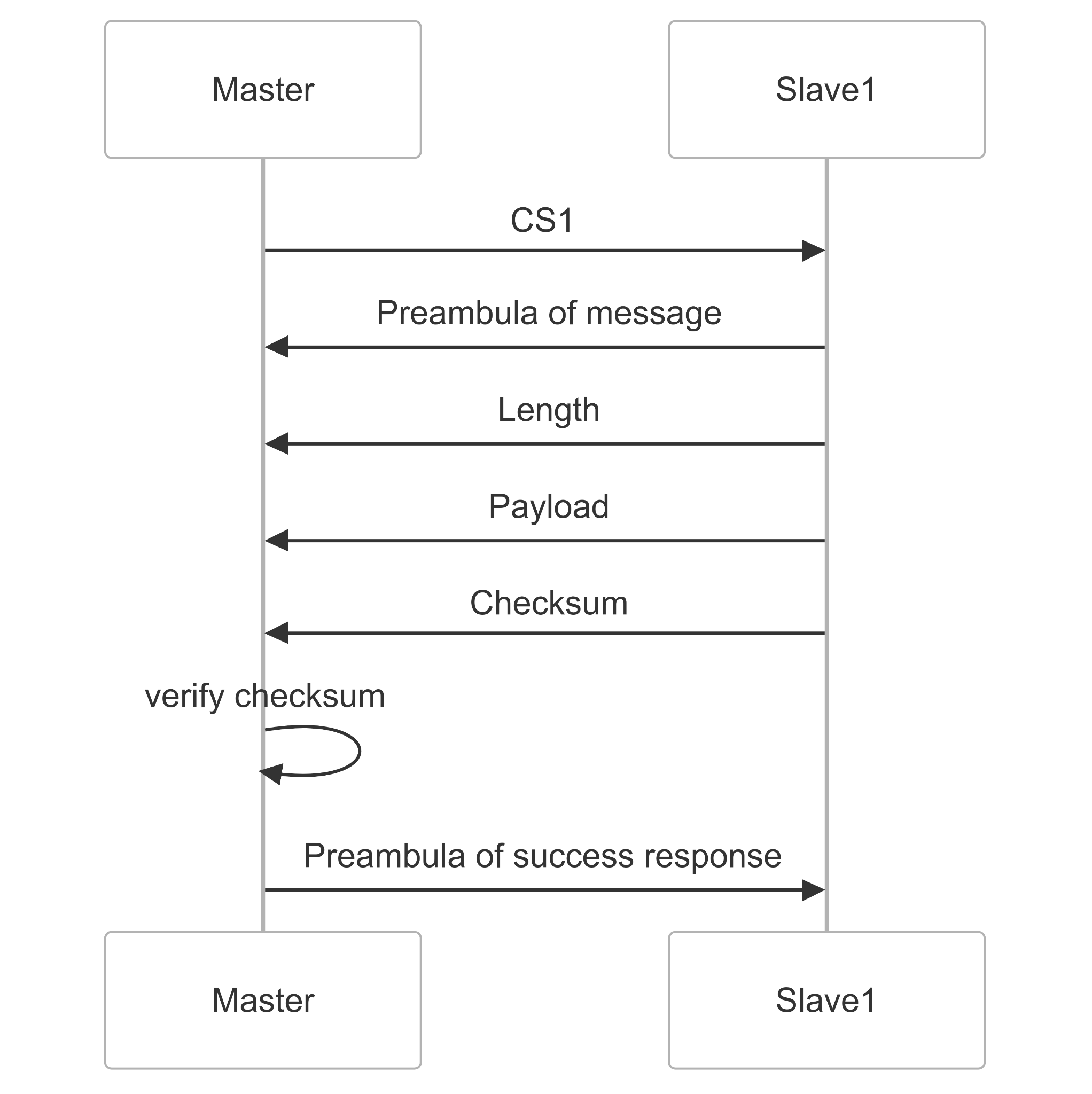
- Master отправляет преамбулу.

- Slave принимает преамбулу и начинает прослушивание.

- Slave считывает длину передаваемых данных и затем принимает сами данные.

- Slave проверяет контрольную сумму и отправляет ответ с подтверждением или требованием повторно отправить пакет.

2. Передача данных от Slave к Master

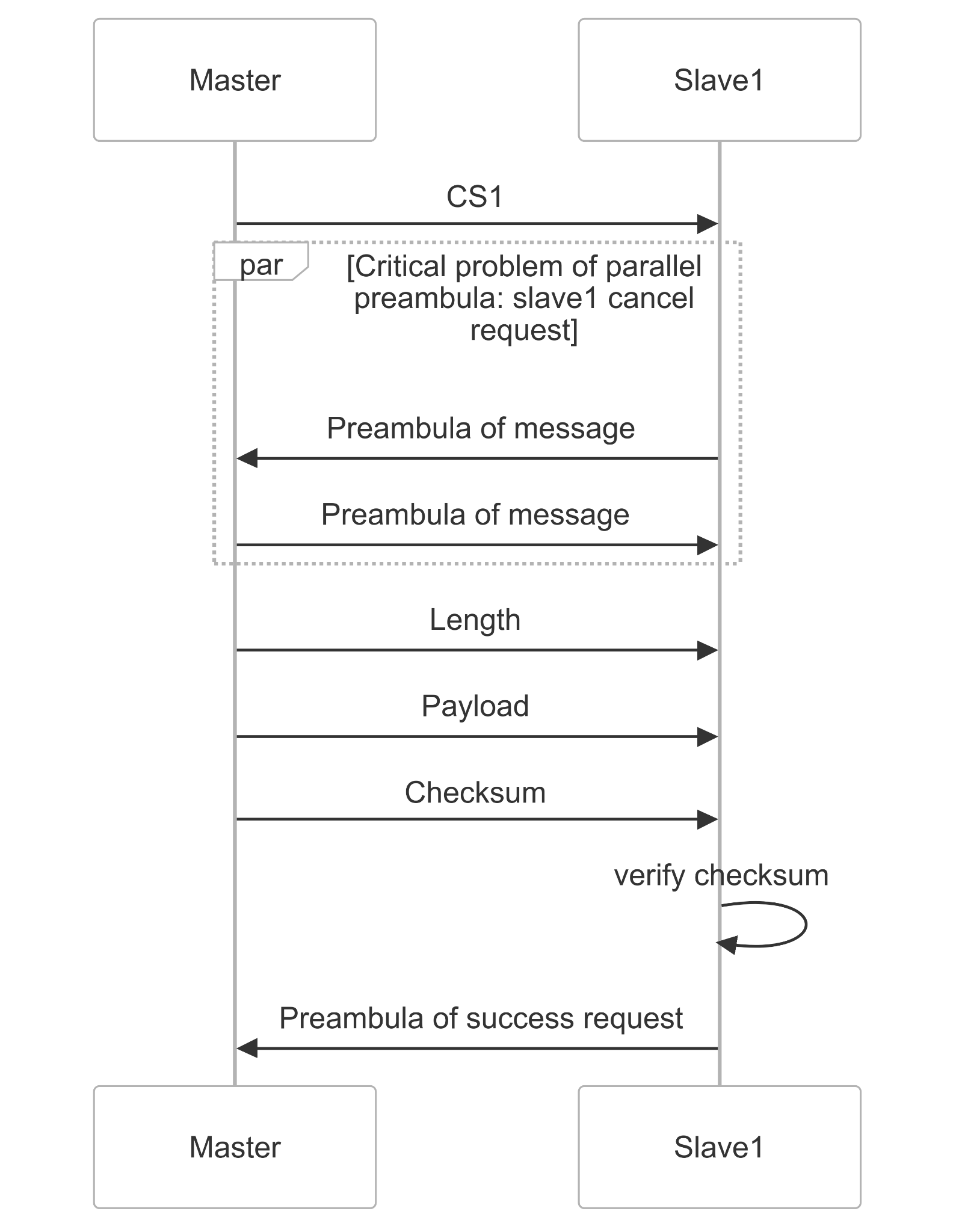


- Master выбирает целевое Slave-устройство с помощью сигнала CS (Chip Select).

- Master ожидает преамбулу от Slave.

- После получения преамбулы взаимодействие продолжается по аналогии с передачей данных от Master.

3. Коллизия при обмене (Master ↔ Slave)



- Master выбирает целевое Slave-устройство с помощью сигнала CS (Chip Select).

- Оба устройства отправляют первые байты преамбулы.

- Если Slave обнаруживает преамбулу, он прерывает вывод данных в Master.

- Slave переходит в режим прослушивания данных от Master, продолжается по аналогии с предыдущими вариантами.

**Вывод**

В ходе выполнения домашнего задания мы спроектировали собственный синхронный дуплексный интерфейс ввода-вывода с 6 линиями связи, описали распиновку его разъема, топологии и способы подключения с помощью него устройств, разработали формат пакета передаваемых данных канального уровня, реализовали адресацию, синхронизацию и надежность передачи данных, изобразили протокол передачи одного байта, рассчитали эффективную пропускную способность интерфейса, определили сценарии и прикладные области использования интерфейса, а также описали протокол транспортного уровня, который позволяет бороться с помехами и потерей пакетов.